NASA

National Aeronautics and Space Administration

EARTH RESOURCES LABORATORY

SCIENTIFIC ANALYSIS SOFTWARE ON THE NASA-ERL COMPUTER SYSTEMS

REPORT NO. 185 JUNE 1980

(NASA-TM-89736) SCIENTIFIC ANALYSIS SOFTWARE ON THE NASA-ERL COMPUTER SYSTEMS (NASA) 64 p

N88-70950

Unclas
00/61 0136021

NATIONAL SPACE TECHNOLOGY LABORATORIES

Mailing Address:

Earth Resources Laboratory
National Space Technology Laboratories
NSTL Station, MS 39529

SCIENTIFIC ANALYSIS SOFTWARE ON THE NASA-ERL COMPUTER SYSTEMS

by

Bobby G. Junkin

Report No. 185

June 1980

SCIENTIFIC ANALYSIS SOFTWARE ON THE

NASA-ERL COMPUTER SYSTEMS

By Bobby G. Junkin*

SUMMARY

Details on the development, usage, and characteristics of 355 scientific subroutines received from the Interdata Users Group are presented in this writeup. These routines, referred to as SSP's (Scientific Subroutine Packages), have been implemented on the NASA Earth Resources Laboratory's Interdata 832 computer system. Program listings and detailed documentation are available for all of the SSP subroutines. For ease of reference, the SSP subroutines are grouped as mathematical or statistical. A further detailed reference is provided by subgroups under these two general categories.

*Earth Resources Laboratory, NASA-NSTL, NSTL Station, MS 39529

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

INTRODUCTION

This report provides details on the development, usage and characteristics of 355 scientific subroutines received from the Interdata Users Group (IUG) (Reference 1). These routines have been implemented on the NASA Earth Resources Laboratory's Interdata 832 computer system. Since they are input/output free computational building blocks, the user must furnish the necessary input/output and other options for his total problem solution.

The various SSP subroutines are categorized as mathematical or statistical and then subgrouped under these categories for ease of referencing. The FORTRAN CALL names and subroutine functions are also included.

Program listings and detailed documentation of the arguments in the FORTRAN CALL statements are available for all of the subroutines. Sample documentation for one of the subroutines is included. Detailed documentation for the remaining subroutines has not been included herein since to do so would require approximately 355 additional pages. A user can extract the appropriate subroutine documentation from Volume P001 via the SSP.CM disk file by routing the printout to the line printer on the Interdata 832.

Additional information is provided on subroutine implementation and usage. Function subprograms, matrix considerations and program and double precision considerations are also discussed.

FUNCTIONAL CATEGORIES

To provide a functional reference for the user, all routines are grouped as mathematical or statistical as follows:

Statistics

- o Probit analysis
- o Analysis of variance (factorial design)
- o Correlation analysis
- o Multiple linear regression
- o Stepwise regression
- o Polynomial regression
- o Canonical correlation
- o Factor analysis (principal components, varimax)
- o Discriminant analysis (many groups)
- o Time series analysis
- o Data screening and analysis
- o Nonparametric tests
- o Distribution functions

Mathematics

- o Inversion
- o Eigenvalues and eigenvectors
- o Simultaneous linear algebraic equations
- o Transpositions
- o Matrix arithmetic (addition, product, etc.)
- o Matrix partitioning

- Matrix tabulation and sorting of rows or columns
- Elementary operations on rows or columns of matrices
- Matrix factorization
- Integration and differentiation of given or tabulated functions
- Solution of systems of first-order differential equations
- Fourier analysis of given or tabulated functions
- Bessel and modified Bessel function evaluation
- Gamma function evaluation
- Jacobian elliptic functions
- Elliptic, exponential, sine cosine, Fresnel integrals
- Finding real roots of a given function
- Finding real and complex roots of a real polynomial
- Polynomial arithmetic (addition, division, etc.)
- Polynomial evaluation, integration, differentiation
- Chebyshev, Hermite, Laguerre, Legendre polynomials
- Minimum of a function
- Approximation, interpolation, and table construction

These groups are then subgrouped as follows:

A. Statistics

.1 Group 1

- Data Screening
- Correlation and Regression
- Analysis of Variance
- Discriminant Analysis
- Factor Analysis

A.2 Group 2

- Time Series
- Nonparametric Statistics

A.3 Group 3

- Generation of Random Variates (Distribution Functions)
- Elementary Statistics and Miscellany

B. Mathematics

B.1 Group 1 : Matrices

- Matrices : Storage
- Matrices : Operations
- Matrices : Special Operations

B.2 Group 2 : Differential Equations

- Numerical Differentiation
- Ordinary Differential Equations
- Numerical Quadrature

B.3 Group 3 : Interpolation, Approximation, Smoothing

B.4 Group 4 : Nonlinear Equations

- Roots
- Extremum of Functions
- Permutations
- Sequences

B.5 Group 5 : Special Functions

B.6 Group 6 : Polynomials

- Operations
- Roots
- Special Types

The individual subroutine names and functions within each group are given in Appendix A.

IMPLEMENTATION ON NASA/ERL INTERDATA 832

All of the IUG-SSP routines have been compiled with the FORTRAN VII optimizing compiler. A tabulation of the number of words in core that the routine occupies when loaded is given in Appendix B, Tables I-XXI. These routines have been placed in a library file named SSP.LIB on volume MT32. They are accessible via FORTRAN CALL statements and can be included in the user task at task building time via the EDIT MT32:SSP.LIB/G statement. This will cause only the referenced routines to be extracted and loaded from the library.

The individual subroutines contain documentation embedded within the comments section. For example, the subroutine NROOT contains comments pertaining to the purpose, usage, parameter descriptions, remarks, subroutines and function subprograms required, and the analysis method. This example is shown in Appendix C.

USAGE OF SSP SUBROUTINES

General

The standard FORTRAN CALL statement is used to access the various subroutines. All data is transmitted via the arguments in the CALL statement. Thus, the routines do not cause input or output directly. The user must define by DIMENSION statements all matrices to be operated on by SSP subroutines as well as those matrices utilized in his program. The SSP subroutines are no different from user-supplied subroutines, with the exception that the dimensional areas in the SSP subroutines are not required to be the same as those in the calling program.

Sequences of Subroutines

Many of the functional categories have been programmed as a sequence of subroutines. An example of this is factor analysis. This statistical procedure is a method of analyzing the inter-correlations within a set of variables. It determines whether the variance in the original set of variables can be accounted for by a smaller number of basic factors. Factor analysis is accomplished by calling the five subroutines in sequence:

1. CORRE - determine mean, standard deviation and correlation matrix.
2. EIGEN - computes eigenvalues and eigenvectors.
3. TRACE - selects eigenvalues.
4. LOAD - computes factor matrix.
5. VARMX - rotates factor matrix.

Several of these routines are also used by other routines.

CALL Statement

The CALL statement transfers control to a particular subroutine and replaces the dummy variables in that subroutine with the value of the arguments appearing in the statement list. If the argument is an array or function subprogram name then the address of the array or function subprogram is transmitted to the called subroutine.

The arguments in a CALL statement must agree in order, number and type with the corresponding arguments in the subroutine. A number may be passed to a subroutine either as a variable name or a constant in the argument list.

Function Subprograms

Several of the subroutines require the name of a user function subprogram as part of the argument list in the CALL statement. The function name appearing in the argument list must then appear in an EXTERNAL statement at the beginning of that program.

For example, the subroutine RK2 integrates a function supplied by the user. If the function is of the form:

$$dy/dx = Ax + By$$

then the function subprogram is:

```
FUNCTION HELP (X,Y)
A = 5.
B = 10.
HELP = A * X + B * Y
RETURN
END
```

Then the main program would appear as follows:

```
EXTERNAL HELP
    .
    .
    .
CALL RK2 (HELP,...)
    .
    .
    .
RETURN
END
```

The subroutine RK2 goes to the function HELP each time it requires a value for the derivative. The dummy function name FUN in RK2 is replaced by the name HELP during execution of the subroutine.

Matrix Considerations

Dimensions

The matrix subroutines do not contain fixed maximum dimensions for data arrays named in their calling sequence. The vector storage approach, wherein each column is followed in storage by the next column, is used to implement this variable dimension capability. This form of storage is the same as two-dimensional storage when the number of rows and columns in the matrix are the same as those in the user's dimension statement. If the matrix is smaller than the dimensional area, then the storage methods are not the same.

As an example, consider the data storage of a 3 x 3 array of numbers in a 5 x 5 dimensional area. For the double subscripted variables, the nine numbers will appear as shown in Figure 1.

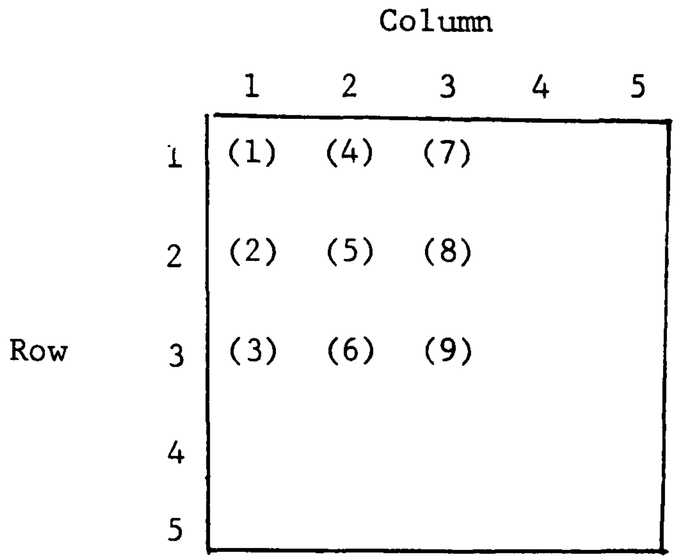

Figure 1.- Double-subscripted data storage.

In this example, sequential core locations will contain data elements 1 to 3, 2 blank locations, data elements 4 to 6, 2 blank locations, etc. The vector storage is illustrated in Figure 2. The matrix subroutines assume the data are stored in this manner. If the user's

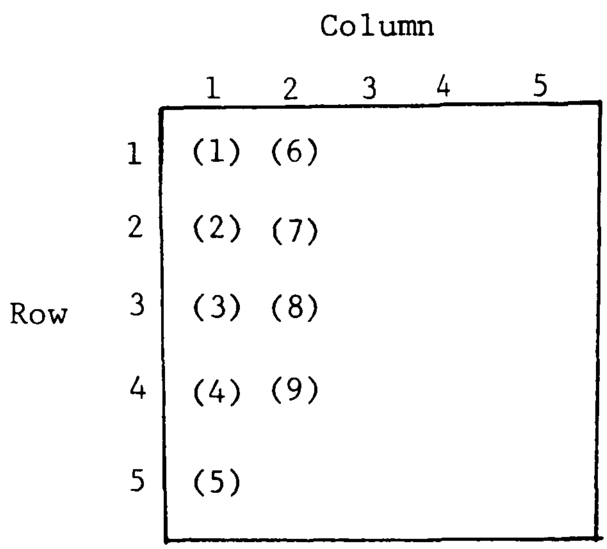

Figure 2.- Vector storage.

dimensional area is larger than the arrays then he must store his data in the vector manner. The SSP subroutine ARRAY is available to accomplish this. In addition, the routine LOC can be used to reference data elements stored in the vector manner.

Modes of Storage

The three modes of storage for using the matrix subroutines in the vector manner are general, symmetric, and diagonal. The general mode is one in which all elements of the matrix are in storage. The symmetric mode uses only the upper triangular portion of the matrix and the diagonal mode uses only the diagonal elements. These modes are summarized in Table 1.

TABLE 1.-SUMMARY OF STORAGE MODES

Type Matrix Storage	Order of Matrix	Storage Locations Required	Storage Mode Code*
General	P x Q	P x Q	0
Symmetric	P	P(P+1)/2	1
Diagonal	P	P	2

*This parameter code is input to the subroutine LOC.

Thus, if the programmer wishes to use SSP subroutines on matrix A, which is general, matrix B, which is symmetric, and matrix C, which is diagonal, and all matrices are 10 by 10 or smaller, the dimension statement in his program can be:

```
DIMENSION A(100), B(55), C(10)
```

It can be seen that considerable savings in data storage can be realized for special matrix forms.

Element References

The subroutine LOC is used to reference matrix elements stored in the vector manner and requires as input the storage mode code in Table XXIII. The calling sequence is:

```
CALL LOC (I,J,K,P,Q,MS)
```

where:

- I - row element reference
- J - column element reference
- K - subscript referring to desired element
- P - number of rows
- Q - number of columns
- MS - storage mode code

If reference is required to the element at row I and column J of matrix A whose dimensions are P by Q, and if the storage mode code is MS, a CALL to the LOC subroutine as shown above will result in the computation of the subscript K such that A(K) is the desired element. The parameters represented by I, J, P, Q, MS can be either integer variables or integer constants. The parameter represented by K is an integer variable. Note that the user must dimention the array A as a single-subscripted variable to meet the restrictions of some FORTRAN systems.

As an example, if reference is required to the row 3 and column 2 element of the 4 x 4 symmetric matrix in Figure 3, then the call statement would be:

CALL LOC (3,2,K,4,4,1)

The value of K returned by LOC will be 5, since the value (6) is found in the fifth storage element location.

(1)	(2)	(3)	(4)
(2)	(5)	(6)	(7)
(3)	(6)	(8)	(9)
(4)	(7)	(9)	(10)

Matrix

Storage	Column
(1)	1
(2)	2
(5)	
(3)	3
(6)	
(8)	
(4)	4
(7)	
(9)	
(10)	

Storage

Figure 3.- Symmetric matrix storage mode.

(1)	0	0	0
0	(2)	0	0
0	0	(3)	0
0	0	0	(4)

Matrix

Storage	Column
(1)	1
(2)	2
(3)	3
(4)	4

Storage

Figure 4.- Diagonal matrix storage mode.

If the storage mode code is for a symmetric matrix where only the upper triangular portion is retained in storage, and if I and J refer to an element in the lower triangular portion, K will contain the subscript for the corresponding element in the retained upper triangle. Thus if the user wanted the element in row 4, column 2 of the matrix shown in Figure 3 and the array was stored as shown, the statement:

```
CALL LOC (4, 2, K, 4, 4, 1)
```

would result in K having the value of 8; that is, the eighth element. If a matrix is stored as shown in Figure 4 (storage mode 2), and LOC is used to compute the subscript for an off-diagonal element (I not equal to J), the result in K will be zero. This is due to the fact that the element does not exist in storage. In this situation, the user must not utilize K as a subscript. Following is an illustration of how to take care of this condition and also handle the case where the current storage mode is unknown.

If the user wishes to set a variable X equal to the element in the third row and fourth column of a 10 by 10 array named A, for either a symmetric, diagonal, or general matrix, the required program can be implemented for any storage mode MS as follows:

```
      CALL LOC (3, 4, K, 10, 10, MS)
      X = 0.0
      IF (K) 20, 30, 20
   20 X = A(K)
   30 - - - -
```

(MS is assumed to have been set a 0, 1, or 2 at some earlier point in the program.) This sequence will then set the proper value for X, given any storage mode that may be encountered. The second and third statements take care of the off-diagonal condition for a matrix with a storage mode of 2.

The subroutine LOC can also be used to compute the total length of an array in storage as follows:

```
CALL LOC (P, Q, K, P, Q, MS)
```

For example, if the user has a 3 by 3 matrix whose storage mode is 1, the statement:

```
CALL LOC (3, 3, K, 3, 3, 1)
```

will result in K being set to 6. This references element 3,3 and is also the actual length of the vector in storage.

The information contained in the fifth parameter in the calling sequence for LOC is not actually used in the calculations performed by LOC. It has been included in the calling sequence in case the user wishes to expand LOC to cover other forms of data storage.

Program Considerations

There are certain FORTRAN compiler features on the Interdata which provide for a versatile program development capability. For example, the FORTRAN VII system includes a subroutine multiple entry point. Usage of this capability in effect reduces total storage requirements.

The user can also optimize execution time by implementing routines in assembly language. For example, the LOC subroutine could be implemented in assembly language and decrease the execution time for the matrix routines which call LOC. The function of LOC could also be coded in-line within each subroutine using it and thus avoid the use of repeated calls to LOC and at the same time decrease execution time.

Double Precision

Many of the subroutines have been provided in single and double precision versions. Those that are not double precision contain instructions for converting. FORTRAN double precision statements have been included in these subroutines in the form of comments cards. The removal of the C from column 1 of the appropriate comments cards prior to program compilation allows one to obtain the double precision version of the subroutine. The basis rules pertaining to double precision usage should be adhered to. In addition, any user supplied function named in the CALL statement for a double precision subroutine must be set up as a double precision function. As an example, if the double precision version of RK2 is used, then the function would be set up as follows:

```
DOUBLE PRECISION FUNCTION HELP (X, Y)
DOUBLE PRECISION X, Y
A = 5.
B = 10.
HELP = A*X + B*Y
RETURN
END
```

CONCLUDING REMARKS

Details on the development, usage, and characteristics of 355 mathematical and statistical subroutines received from the Interdata Users Group have been presented. This report serves as a reference guide for potential users of this scientific analysis software capability.

Routine software maintenance, which includes documentation changes, may be required as a result of user feedback information and/or in-house investigations. In addition, new mathematical or statistical subroutines will be implemented in the software library as requirements dictate.

REFERENCE

1. "Scientific Subroutine Package," Interdata Users Group - 320, Oceanport, NJ.

APPENDIX A

Subroutine Names and Functions

This appendix contains a listing of the various subroutines and the particular mathematical or statistical category that each subroutine is included under. The statistical groupings and number in each group are as follows:

A.1 Statistics - Group I/24

A.2 Statistics - Group II/18

A.3 Statistics - Group III/12

Group I pertains to data screening, correlation, regression, and variance and factor analysis. Group II is concerned with time series and nonparametric statistics. Group III includes distribution functions and elementary statistics.

The mathematical groupings and number in each group are as follows:

B.1 Mathematics - Group I/74

B.2 Mathematics - Group II/96

B.3 Mathematics - Group III/38

B.4 Mathematics - Group IV/16

B.5 Mathematics - Group V/20

B.6 Mathematics - Group VI/56

Groups I and II include matrices and differential equations. Group III pertains to interpolation, approximation and smoothing. Nonlinear equation routines are summarized in Group IV. Special functions are given in Group V. Finally, Group VI covers polynomial routines.

A.1 STATISTICS - Group I

A.1.1 Data Screening

TALLY - totals, means, standard deviations, minimums, and maximums

BOUND - selection of observations within bounds

SUBST - subset selection from observation matrix

ABSNT - detection of missing data

TAB1 - tabulation of data (1 variable)

TAB2 - tabulation of data (2 variables)

SUBMX - building of subset matrix

A.1.2 Correlation and Regression

CORRE - means, standard deviations, and correlations

MISR - means, standard deviations, third and fourth moments. Correlations, simple regression coefficients and their standard errors; considers that data may be missing

ORDER - rearrangement of intercorrelations

MULTR - multiple linear regression

GDATA - data matrix generation for polynomial regression

STPRG - stepwise multiple linear regression

PROBT - probit analysis

CANOR - canonical correlation

NROOT - eigenvalues and eigenvectors of a special non-symmetric matrix

A.1.3 Analysis of Variance

AVDAT - data storage allocation

AVCAL - data operation for factorial design

MEANQ - mean square operation

A.1.4 Discriminant Analysis

DMATX - means and dispersion matrix

DISCR - discriminant functions

A.1.5 Factor Analysis

TRACE - cumulative percentage of eigenvalues

LOAD - factor loading

VARMX - varimax rotation

A.2 STATISTICS - Group II

A.2.1 Time Series

AUTO - autocovariances

CROSS - crosscovariances

SMO - application of filter coefficients (weights)

EXSMO - triple exponential smoothing

Nonparametric Statistics

KOLMO - Kolmogorov-Smirnov one-sample test

KOLM2 - Kolmogorov-Smirnov two-sample test

SMIRN - Kolmogorov-Smirnov limiting distribution values

CHISQ - χ^2 test for contingency tables

KRANK - Kendall rank correlation

MPAIR - Wilcoxin's signed ranks test

QTEST - Cochran Q-test

RANK - rank observations

SIGNT - sign test

SPANK - Spearman rank correlation

TIE - calculation of ties in ranked observations

TWOAV - Friedman two-way analysis of variance statistic

UTEST - Mann-Whitney U-test

WTEST - Kendall coefficient of concordance

A.3 STATISTICS - Group III

A.3.1 Generation of Random Variantes - Distribution Functions

GAUSS - normal deviates

NDTR - normal distribution function

BDTR - beta distribution

CDTR - χ^2 distribution function

NDTRI - inverse of normal distribution function

A.3.2 Elementary Statistics and Miscellany

MOMEN - first four moments

TTEST - test on population means

BISER - biserial correlation coefficient

PHI - phi coefficient

POINT - point-biserial correlation coefficient

TETRA - tetrachoric correlation coefficient

SRATE - survival rates

B.1 MATHEMATICS - Group I : Matrices

B.1.1 Matrices: Storage

MCPY - matrix copy

RCPY - copy row of matrix into vector

CCPY - copy column of matrix into vector

DCPY - copy diagonal of matrix into vector

XCPY - copy submatrix from given matrix

MSTR - storage conversion

LOC - location in compressed-stored matrix

CONVT- single-precision/double-precision conversion

ARRAY- vector storage/double-dimensional storage conversion

B.1.2 Matrices: Operations

GMADD- add two general matrices

GMSUB- subtract two general matrices

GMPRD- product of two general matrices

GMTRA- transpose of a general matrix

GTPRD- transpose product of two general matrices

MADD - add two matrices

MSUB - subtract two matrices

MPRD - matrix product (row into column)

MTRA - transpose a matrix

TPRD - transpose product

MATA - transpose product of matrix by itself

SADD - add scalar to matrix

SSUB - subtract scalar from a matrix

SMPY - matrix multiplied by a scalar

SDIV - matrix divided by a scalar

SCLA - matrix clear and add scalar

DCLA - replace diagonal with scalar

RADD - add row of one matrix to row of another matrix

CADD - add column of one matrix to column of another matrix

SRMA - scalar multiply row and add to another row

SCMA - scalar multiply column and add to another column

RINT - interchange two rows

CINT - interchange two columns

RSUM - sum the rows of a matrix

CSUM - sum the columns of a matrix

RTAB - tabulate the rows of a matrix

CTAB - tabulate the columns of a matrix

RSRT - sort matrix rows

CSRT - sort matrix columns

RCUT - partition by row

CCUT - partition by column

RTIE - adjoin two matrices by row

CTIE - adjoin two matrices by column

MPRC, DMPRC - permute rows or columns

MFUN - matrix transformation by a function

RECP - reciprocal function for MFUN

B.1.3 Matrices: Special Operations

RSLMC - solution of system of linear equations

SIMQ - solution of a set of simultaneous linear equations

MINV - matrix inversion

FACTR - factorization of matrix

EIGEN - eigenvalues and eigenvectors of real symmetric matrix

HSBG - reduce a real matrix into upper almost triangular form

ATEIG- eigenvalues of real almost triangular matrix

SINV, DSINV - symmetric positive definite matrix inversion

GELB, DGELB - solve a system of simultaneous linear equations with coefficient matrix of baud structures

GELG, DGELG - solve a general system of simultaneous linear equations

GELS, DGELS - solve a system of simultaneous linear equations with symmetric coefficient matrix

LLSQ, DLLSQ - linear least squares problem

MCHB, DMCHB - factorization of matrix using Cholesky's square root method

MFGR, DMFGR - calculate triangle factors or matrix using Gauss elimination technique

MFSD, DMFSD - factor symmetric positive definite matrix

MFSS, DMFSS - factor symmetric positive semi-definite matrix

MLSS, DMLSS - least squares solution of system of simultaneous linear equations

MTDS, DMTDS - solution of system of equations with symmetric positive definite coefficient matrix

B.2 MATHEMATICS - Group II: Differential Equations

B.2.1 Numerical Differentiation

DGT3, DDGT3 - differentiation of a tabulated function by parabolic interpolation

DET3, DDET3 - differentiation of an equidistantly tabulated function

DET5, DDET5 - differentiation of an equidistantly tabulated function

DCAR, DDCAR - derivative of a function at the center of an interval

DBAR, DDBAR - derivative of a function at the border of an interval

B.2.2 Ordinary Differential Equations

RK1 - solution of first-order differential equation by Runge-Kutta method

RK2 - tabulated solution of first-order differential equation by Runge-Kutta method

RKGS, DRKGS - solution of system of first-order ordinary differential equations with given initial values by the Runge-Kutta method

HPCG, DHPCG - solution of general system of first-order ordinary differential equations with given initial values by Hamming's modified predictor-corrector method

HPCL, DHPCL - solution of linear system of first-order ordinary differential equations with given initial values by Hamming's modified predictor-corrector method

LBVP, DLBVP - solution of linear first-order ordinary differential equations with linear boundary conditions by method of adjoint equations

B.2.3 Numerical Quadrature

QTFG, DQTFG - integration of monotonically tabulated function by trapezoidal rule

QTFE, DQTFE - integration of equidistantly tabulated function by trapezoidal rule

QSF, DQSF - integration of equidistantly tabulated function by Simpson's rule.

QHFG, DQHFG - integration of monotonically tabulated function with first derivative by Hermitian formula of first order

QHFE, DQHFE - integration of equidistantly tabulated function with first derivative by Hermitian formula of first order.

QHSG, DQHSG - integration of monotonically tabulated function with first and second derivatives by Hermitian formula of first order

QHSE, DQHSE - integration of equidistantly tabulated function with first and second derivatives by Hermitian formula of second order

QATR, DQATR - integration of a given function by trapezoidal rule together with Romberg's extrapolation method

QG2-QG10, DQG4-DQG32 - integration of a given function by Gaussian quadrature formulas

QL2-QL10, DQL4-DQL32 (4,8,12,16 24,32 only) - integration of a given function by Gaussian-Laguerre quadrature formulas

QH2-QH10, DQH8-DQH64 (8,16,24,32, 48,64 only) - integration of a given function by Gaussian-Hermite quadrature formulas

QA2-QA10, DQA4-DQA32 - integration of a given function by associated Gaussian-Laguerre quadrature formulas

B.3 MATHEMATICS - Group III - INTERPOLATION, APPROXIMATION, SMOOTHING

ALI, DALI - Aitken-Lagrange interpolation

AHI, DAHI - Aitken-Hermite interpolation

ACFI, DACFI - continued fraction interpolation

ATSG, DATSG - table selection out of a general table

ATSM, DATSM - table selection out of a monotonic table

ATSE, DATSE - table selection out of an equidistant table

SG13, DSG13 - local least-squares smoothing of tabulated functions

SE13, DSE13 - local least-squares smoothing of equidistantly tabulated functions

SE15, DSE15 - local least-squares smoothing of equidistantly tabulated functions

SE35, DSE35 - local least-squares smoothing of equidistantly tabulated functions

APFS, DAPFS - solve normal equations for least-squares fit

APCH, DAPCH - least-squares polynomial approximation

ARAT, DARAT - rational least-squares approximation

FRAT, DFRAT - used by ARAT, DARAT

APLL, DAPLL - linear least-squares approximation

FORIF - Fourier analysis of a given function

FORIT - Fourier analysis of a tabulated function

HARM, DHARM - complex three-dimensional analysis

RHARM, DRHARM- real one-dimensional analysis

APMM, DAPMM - linear Chebyshev approximation over a discrete range

B.4 MATHEMATICS - Group IV: Nonlinear Equations

B.4.1 Roots

RTWI, DRTWI - refine estimate of root by Wegstein's iteration

RTMI, DRTMI - determine root within a range by Mueller's iteration

RTNI, DRTNI - refine estimate of root by Newton's iteration

B.4.2 Extremum of Functions

FMFP, DFMFP - unconstrained minimum of a function of several variables - Davidon method

FMCG, DFMCG - unconstrained minimum of a function of several variables - conjugate gradient method

B.4.3 Permutations

PPRCN - composition of permutations

PERM - operations with permutations and transpositions

B.4.4 Sequences

TEAS, DTEAS - limit of a given sequence

TEUL, DTEUL - sum of a given function sequence

B.5 MATHEMATICS - Group V: Special Functions

GMMMA - gamma function

DLGAM - natural log of gamma function

BESJ - J Bessel function

BESK - K Bessel function

BESY - Y Bessel function

INUE - modified Bessel functions for orders 1 to N

IO - modified Bessel function of order Ø

EXPI - exponential integral

SICI - sine cosine integral

CS - Fresnel integrals

CEL1, DCEL1 - complete elliptic integral of the first kind

CEL2, DCEL2 - complete elliptic integral of the second kind

ELI1, DELI1 - generalized elliptic integral of the first kind

ELI2, DELI2 - generalized elliptic integral of the second kind

JELF, DJELF - Jacobian elliptic functions

B.6 MATHEMATICS - Group VI: Polynomials

B.6.1 Operations

PADD - add two polynomials

PSUB - subtract one polynomial from another

PCLA - replace one polynomial by another

PADDM- multiply polynomial by constant and add to another polynomial

PMPY - multiply two polynomials

PDIV - divide one polynomial by another

PVAL - value of a polynomial

PVSUB- substitute variable of polynomial by another polynomial

PILD - evaluate polynomial and its first derivative

PDER - derivative of a polynomial

PINT - integral of a polynomial

PQSD - quadratic synthetic division of a polynomial

PCLD - complete linear synthetic division

PGCD - greatest common divisor of two polynomials

PNORM- normalize coefficient vector of polynomial

PECN, DPECN - economization of a polynomial for symmetric range

PECS, DPECS - economization of a polynomial for unsymmetric range

B.6.2 Roots

POLRT - real and complex roots of a real polynomial

PRQD, DPRQD - roots of a real polynomial by QD algorithm with displacement

PRBM, DPRBM - roots of a real polynomial by Bairstow's algorithm

PQFB, DPQFB - determine a quadratic factor of a real polynomial

B.6.3 Special Types

CNP, DCNP - value of N^{th} Chebyshev polynomial

CNPS, DCNPS - value of series expansion in Chebyshev polynomials

TCNP, DTCNP - transform series expansion in Chebyshev polynomials to a polynomial

CSP, DSCP - value of N^{th} shifted Chebyshev polynomial

CSPS, DCSPS - value of series expansion in shifted Chebyshev polynomials

TCSP, DTCSP - transform series expansion in shifted Chebyshev polynomials to a polynomial

HEP, DHEP - value of Hermite polynomial

HEPS, DHEPS - value of series expansion in Hermite polynomials

THEP, DTHEP - transform series expansion in Hermite polynomials to a polynomial

LAP, DLAP - value of Laguerre polynomial

LAPS, DLAPS - value of series expansion in Laguerre polynomials

TLAP, DTLAP - transform series expansion in Laguerre polynomials to a polynomial

LEP, DLEP - value of Legendre polynomial

LEPS, DLEPS - value of series expansion in Legendre polynomials

TLEP, DTLEP - transform a series expansion in Legendre polynomials to a polynomial

APPENDIX B

Subroutine Core Requirements

This appendix contains a listing of each subroutine in the SSP1-SSP15 modules and the SSPA-SSPF modules. The table space and object size for each subroutine is included.

TABLE I. - SSP1 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	CANOR	65K	1	7FC
2	NROOT	53K	6	534
3	PROBT	61K	9	5D0
4	GDATA	51K	14	438
5	LOAD	33K	17	E4
6	MEANQ	51K	19	3C0
7	MISR	63K	23	8C0

Largest table space needed: 65K in program CANOR

Total length of object: 249C bytes (hexadecimal)

TABLE II. - SSP2 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	MULTR	49K	1	3A4
2	ORDER	35K	5	1B8
3	ABSNT	33K	8	F0
4	AVCAL	39K	10	1A4
5	AVDAT	41K	13	264
6	BOUND	37K	16	250
7	CORRE	77K	19	7F8
8	DISCR	71K	24	79C
9	DMATX	43K	28	3A4
10	STPRG	77K	31	8F8
11	SUBMX	33K	38	128
12	SUBST	35K	40	23C
13	TAB1	49K	43	324
14	TAB2	65K	46	634
15	TALLY	41K	50	2EC
16	TRACE	35K	53	120
17	VARMX	69K	55	6F8
18	KOLMO	45K	60	360
19	KOLM2	45K	64	2B8

Largest table space needed: 77K in program CORRE

Total length of object: 46AC Bytes (hexadecimal)

TABLE III.- SSP3 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	KRANK	43K	1	3C4
2	MPAIR	39K	4	314
3	SIGNT	39K	7	37C
4	SPANK	37K	10	2F8
5	TWOAV	37K	13	220
6	UTEST	35K	15	230
7	WTEST	43K	17	2FC
8	QTEST	35K	20	14C
9	AUTO	37K	22	124
10	CHISQ	45K	24	2B4
11	CROSS	37K	27	190
12	EXSMO	37K	29	1D0
13	RANK	35K	31	164
14	SMIRN	33K	33	15C
15	SMO	35K	36	1A0
16	TIE	35K	38	154
17	BDTR	63K	40	95C
18	BISER	43K	45	23C
19	CDTR	61K	48	830

Largest table space needed: 63K in program BDTR

Total length of object: 3598 bytes (hexadecimal)

TABLE IV.- SSP4 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	GAUSS	31K	1	DC
2	TETRA	51K	3	5FC
3	MOMEN	41K	7	258
4	NDTR	31K	10	110
5	NDTRI	33K	12	198
6	PHI	39K	14	280
7	POINT	41K	17	1F8
8	SRATE	43K	20	2FC
9	TTEST	43K	23	33C
10	QATR	39K	26	2D8
11	QA10	33K	29	1EO
12	QA2	31K	31	BO
13	QA3	31K	33	D4
14	QA4	31K	35	100
15	QA5	33K	37	124
16	QA6	33K	39	148
17	QA7	33K	41	16C
18	QA8	33K	43	198
19	QA9	33K	45	1BC
20	QG10	33K	47	21C
21	QG2	31K	49	E8
22	QG3	31K	51	110
23	QG4	33K	53	134
24	QG5	33K	55	164
25	QG6	33K	57	184
26	QG7	33K	59	1AC

Largest table space needed: 51K in program TETRA
Total length of object: 2ECC bytes (hexadecimal)

TABLE V.- SSP5 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	QG8	33K	1	1CC
2	QG9	33K	3	1FC
3	QHFE	33K	5	104
4	QHFG	33K	7	11C
5	QHSE	33K	9	134
6	QHSG	35K	11	154
7	QH10	33K	13	1C4
8	QH2	31K	15	B0
9	QH3	31K	17	D4
10	QH4	31K	19	F8
11	QH5	31K	21	11C
12	QH6	33K	23	13C
13	QH7	33K	25	160
14	QH8	33K	27	180
15	QH9	33K	29	1A4
16	QL10	33K	31	1E0
17	QL2	31K	33	B0
18	QL3	31K	35	D4
19	QL4	31K	37	100
20	QL5	33K	39	124
21	QL6	33K	41	148
22	QL7	33K	43	16C
23	QL8	33K	45	198
24	QL9	33K	47	1BC
25	QSF	51K	49	2A8
26	DQA12	35K	52	2D4
27	DQA16	35K	54	39C
28	DQA24	39K	56	534

TABLE V.- SSP5 MODULE - (CONTINUED)

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
29	DQA32	43K	59	6C4
30	DQG12	35K	62	2C4
31	DQG16	35K	64	36C
32	DQG24	37K	66	4BC
33	DQG32	39K	68	60C

Largest table space needed: 51K in program QSF

Total length of object: 4524 bytes (hexadecimal)

TABLE VI.- SSP6 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	DQHSG	35K	1	184
2	DQH16	35K	3	34C
3	DQH24	37K	5	4AC
4	DQH32	41K	7	60C
5	DQH48	45K	10	8DC
6	DQH64	51K	13	B9C
7	DQL12	35K	17	2D4
8	DQL16	35K	19	39C
9	DQL24	39K	21	534
10	DQL32	43K	24	6C4
11	DQATR	43K	27	384
12	DQA4	31K	30	144
13	DQA8	33K	32	20C
14	DQG4	33K	34	174
15	DQG8	33K	36	21C
16	DQHFE	33K	38	134
17	DQHFG	33K	40	150
18	DQHSE	33K	42	170
19	DQH8	33K	44	1EC
20	DQL4	31K	46	144
21	DQL8	33K	48	20C
22	DQSF	51K	50	2F4
23	DQTFE	33K	53	F0
24	DQTFG	33K	55	104
25	QTFE	33K	57	D0

Largest table space needed: 51K in program DQH64

Total length of object: 5278 bytes (hexadecimal)

TABLE VII.- SSP7 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	QTFG	33K	1	E8
2	DSINV	37K	3	2A0
3	SINV	41K	6	30C
4	GELB	69K	9	80C
5	GELG	57K	14	5CC
6	GELS	57K	18	5F4
7	HSBG	49K	22	3E8
8	LLSQ	75K	25	83C
9	MCHB	59K	30	6F0
10	MFGR	63K	36	6BC
11	MFSD	37K	41	20C

Largest table space needed: 75K in program LLSQ

Total length of object: 363C bytes (hexadecimal)

TABLE VIII.- SSP8 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	MFSS	69K	1	800
2	MINV	53K	6	424
3	MLSS	53K	10	578
4	MTDS	45K	14	470
5	ATEIG	95K	18	B88
6	DGELB	71K	24	858
7	DGELG	59K	29	61C
8	DGELS	59K	33	638
9	DLLSQ	79K	37	898
10	DMCHB	59K	42	660
11	DMFGR	63K	48	710

Largest table space needed: 95K in program ATEIG

Total length of object: 4C48 bytes (hexadecimal)

TABLE IX.- SSP9 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	DMLSS	49K	1	4B0
2	DMFSD	35K	5	1D4
3	DMFSS	69K	8	72C
4	DMTDS	43K	13	3F8
5	EIGEN	59K	17	5F4
6	FACTR	47K	21	45C
7	RSLMC	55K	24	5A4
8	SIMQ	45K	28	364
9	CSRT	37K	31	21C
10	CTAB	35K	34	258
11	MTRA	33K	36	158
12	RTAB	35K	38	258
13	CADD	33K	40	188

Largest table space needed: 69K in program DMFSS

Total length of object: 2EAC bytes (hexadecimal)

TABLE X.- SSP10 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	ALI	35K	1	1CC
2	APCH	47K	4	350
3	APFS	47K	8	398
4	APLL	37K	12	264
5	APMM	103K	15	AEC
6	ATSE	39K	23	20C
7	ATSG	37K	26	1F0
8	ATSM	43K	29	2AC
9	DACFI	47K	32	3A4
10	DAHI	43K	36	284
11	DALI	35K	39	214
12	DAPCH	47K	42	3B0

Largest table space needed: 103K in program APMM

Total length of object: 2898 bytes (hexadecimal)

TABLE XI.- SSP11 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	DAPFS	47K	1	3F0
2	DAPLL	37K	5	278
3	DAPMM	101K	8	A74
4	DSE13	35K	16	170
5	DSE15	39K	18	1B8
6	DHARM	137K	20	144C
7	DSE35	37K	30	1AC
8	DSG13	37K	32	240
9	DATSE	39K	34	234
10	DATSG	37K	37	214
11	DATSM	43K	40	2CC
12	SE13	35K	43	140
13	SE15	39K	43	190

Largest table space needed: 137K in program DHARM

Total length of object: 3620 bytes (hexadecimal)

TABLE XII.- SSP12 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	SE35	37K	1	17C
2	SG13	37K	3	200
3	GMMMA	35K	5	1B0
4	INUE	35K	7	1D8
5	IO	33K	9	184
6	JELF	37K	11	2C0
7	BESJ	41K	14	30C
8	BESK	49K	17	510
9	BESY	43K	20	480
10	CEL1	33K	23	EC
11	CEL2	33K	25	168
12	CS	35K	27	2A0
13	DCEL1	35K	29	144
14	DCEL2	35K	31	1E8
15	DELI1	35K	33	1DC
16	DELI2	39K	35	390
17	DJELF	37K	38	3A0
18	DLGAM	35K	41	228
19	ELI1	35K	43	184
20	ELI2	39K	45	30C
21	EXPI	35K	48	2E8
22	SICI	37K	50	308

Largest table space needed: 49K in program BESK

Total length of object: 35B8 bytes (hexadecimal)

TABLE XIII.- SSP13 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	PGCD	35K	1	1B8
2	DPRBM	63K	3	604
3	PDIV	35K	8	1F8
4	PILD	31K	10	120
5	PRBM	63K	12	5BC
6	PVSUB	37K	17	258
7	PCLA	31K	19	88
8	PCLD	31K	21	A0
9	PADDM	33K	23	118
10	PDER	33K	25	CC
11	PECN	39K	27	1DC
12	PECS	35K	30	1A0
13	PINT	35K	32	C4
14	POLRT	59K	34	608
15	PMPY	33K	38	15C
16	PNORM	31K	40	90
17	PQFB	59K	42	668
18	PQSD	31K	47	C8

Largest table space needed: 63K in program DPRBM

Total length of object: 2958 bytes (hexadecimal)

TABLE XIV.- SSP14 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	PRQD	99K	1	BB0
2	PSUB	33K	9	110
3	PVAL	31K	11	90
4	DPECN	41K	13	21C
5	DPECS	39K	16	200
6	DPQFB	59K	19	6E4
7	DPRQD	99K	24	C10
8	PADD	33K	32	100
9	HEP	35K	34	C0
10	HEPS	35K	36	E4
11	LAP	35K	38	D8
12	LAPS	35K	40	EC
13	LEP	35K	42	C8
14	LEPS	35K	44	E8
15	CNP	35K	46	A4
16	CNPS	35K	48	E4
17	CSP	35K	50	B8
18	CSPS	35K	52	F0
19	DLAP	35K	54	F8
20	DLAPS	35K	56	110
21	DLEP	35K	58	F0
22	DLEPS	35K	60	110
23	DHEP	35K	62	DC

Largest table space needed: 99K in program PRQD

Total length of object: 328C bytes (hexadecimal)

TABLE XV.- SSP15 MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	DHEPS	35K	1	104
2	DCNP	35K	3	C0
3	DCNPS	35K	5	104
4	DCSP	35K	7	E0
5	DCSPS	35K	9	114
6	DTCNP	41K	11	1E4
7	DTCSP	39K	14	1F4
8	DTHEP	41K	17	20C
9	DTLAP	41K	20	294
10	DTLEP	37K	23	264
11	TCNP	39K	26	19C
12	TCSP	41K	28	1AC
13	THEP	41K	31	1C0
14	TLAP	43K	34	28C
15	TLEP	41K	37	200

Largest table space needed: 43K in program TLAP

Total length of object: 192C bytes (hexadecimal)

TABLE XVI.- SSPA MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	CCPY	33K	1	164
2	CCUT	35K	3	1CC
3	CSUM	33K	5	17C
4	CTIE	35K	7	1E8
5	DCLA	31K	9	F8
6	DCPY	33K	11	FC
7	MADD	39K	13	2B8
8	MATA	35K	15	274
9	MCPY	31K	17	F4
10	MFUN	33K	19	134
11	MPRD	39K	21	2DC
12	MSTR	33K	23	1B0
13	MSUB	39K	25	2B8
14	RADD	33K	27	17C
15	RCPY	33K	29	164
16	RCUT	35K	31	1CC
17	RSRT	37K	33	240
18	RSUM	33K	36	17C
19	RTIE	35K	38	214
20	SADD	33K	40	114
21	SCLA	31K	42	FC
22	SDIV	33K	44	120
23	SMPY	33K	46	114
24	SSUB	33K	48	114

TABLE XVI.- SSPA MODULE (CONTINUED)

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
25	TPRD	39K	50	2DC
26	XCPY	33K	52	1C0
27	GMADD	31K	54	AC
28	GMPRD	33K	56	188
29	GMSUB	31K	58	AC
30	GMTRA	33K	60	C0
31	GTPRD	33K	62	188
32	LOC	33K	64	D8

Largest table space needed: 39K in program MADD

Total length of object: 3190 bytes (hexadecimal)

TABLE XVII.- SSPB MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	MPRC	35K	1	1B0
2	ARRAY	35K	4	16C
3	CINT	33K	6	C0
4	CONVT	35K	8	13C
5	DMPRC	35K	10	1BC
6	RECP	31K	13	88
7	RINT	33K	15	C8
8	SCMA	33K	17	D8
9	SRMA	33K	19	E0
10	DLBVP	187K	21	1E18
11	LBVP	187K	32	1D54

Largest table space needed: 187K in program DLBVP

Total length of object: 4548 bytes (hexadecimal)

TABLE XVIII.- SSPC MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	HPCG	133K	1	14B8
2	HPCL	141K	8	1530
3	DBAR	39K	16	300
4	DCAR	39K	19	32C
5	DDBAR	41K	22	3E8
6	DDCAR	41K	25	3A8
7	DDGT3	37K	28	1CC
8	DDET3	35K	31	180
9	DDET5	39K	33	2C8
10	DET3	35K	36	150
11	DET5	39K	38	26C
12	DGT3	37K	40	1A0

Largest table space needed: 141K in program HPCL

Total length of object: 4314 bytes (hexadecimal)

TABLE XIX.- SSPD MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	DHPCG	133K	1	1590
2	DHPCL	141K	8	15E4
3	DRKGS	65K	16	988
4	RKGS	65K	22	904
5	RK1	39K	27	3C4
6	RK2	33K	31	204
7	PPRCN	35K	33	1CC
8	PERM	39K	35	1EC
9	DFMCG	67K	38	7B8
10	DFMFP	83K	44	9C8

Largest table space needed: 141K in program DHPCL

Total length of object: 5900 bytes (hexadecimal)

TABLE XX.- SSPE MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	DRTMI	45K	1	41C
2	DRTNI	35K	5	230
3	DRTWI	35K	8	1F4
4	DTEAS	57K	11	4E0
5	DTEUL	37K	16	27C
6	FMCG	65K	19	71C
7	FMFP	81K	25	924
8	RTMI	43K	31	3B0
9	RTNI	33K	35	1BC
10	RTWI	33K	38	198
11	TEAS	57K	41	45C
12	TEUL	37K	45	1EC

Largest table space needed: 81K in program FMFP

Total length of object: 2D28 bytes (hexadecimal)

TABLE XXI.- SSPF MODULE

Batch Number	PGM Name	Table Space	Page Number	Obj. Size
1	ARAT	67K	1	834
2	DARAT	69K	6	8CC
3	DFRAT	39K	11	1FC
4	DRHARM	53K	14	444
5	FRAT	39K	17	1D4
6	RHARM	51K	20	3F4
7	FORIF	37K	23	230
8	FORIT	37K	26	1A8
9	HARM	137K	29	1400
10	ACFI	45K	39	318
11	AHI	43K	43	23C

Largest table space needed: 137K in program HARM

Total length of object: 3A34 bytes (hexadecimal)

APPENDIX C

Subroutine Documentation Example

SUBROUTINE NROOT

PURPOSE:

Compute eigenvalues and eigenvectors of a real nonsymmetric matrix of the form B-inverse times A. This subroutine is normally called by subroutine CANOR in performing a canonical correlation analysis

USAGE:

Call NROOT (M,A,B,XL,X)

DESCRIPTION OF PARAMETERS:

M - Order of square matrices A, B, and X.

A - Input matrix (M x M).

B - Input matrix (M x M).

XL - Output vector of length M containing eigenvalues of B-inverse times A.

X - Output matrix (M x M) containing eigenvectors column-wise.

REMARKS:

None

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:

EIGEN

METHOD:

Refer to W. W. Cooley and P. R. Lohnes, "Multivariate Procedures for the Behavioral Sciences," John Wiley and Sons, 1962, Chapter 3.